

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

Zadání bakalářské práce

Student:

Jakub Ondruš

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: ZWA Company spol. s r.o.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.


Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Pavel Krömer, Ph.D.**


Konzultant bakalářské práce: Bc. Pavel Šrom

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012


doc. Dr. Ing. Eduard Sojka
vedoucí katedry




prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 4. května 2012



.....

Na tomto místě bych chtěl poděkovat vedení ZWA Company spol. s.r.o. za možnost absolvovat bakalářskou praxi v této firmě. Také bych rád poděkoval konzultantovi bakalářské práce Bc. Pavlu Šromovi za pomoc při řešení zadaných úkolů, za seznámení s chodem firmy a odborné rady. Děkuji také svému vedoucímu bakalářské práce Ing. Pavlu Krömerovi, Ph.D. za připomínky vedoucí ke zlepšení bakalářské práce.

Abstrakt

Tato bakalářská práce popisuje průběh odborné praxe ve firmě ZWA Company spol. s.r.o. V první části je představení firmy ZWA Company spol. s.r.o. a zařazení studenta v rámci odborné praxe. V další části jsou popsány zadání úkolů a postup jejich řešení včetně použitých technologií. V předposlední části jsou uvedeny znalosti a dovednosti, které student při vykonávání odborné praxe využil, avšak jsou zde uvedeny i scházející znalosti a dovednosti. Závěrečná část se věnuje dosaženým výsledkům a celkovému zhodnocení odborné praxe.

Klíčová slova: .NET, XML, WinForms, XPath, HTML,C#, Visual studio 2010

Abstract

This thesis describes the course of professional practice in ZWA Company Ltd. The first part shows ZWA Company Ltd. and the inclusion of student in professional practice. The next section describes the specified tasks and their solution, including technology used. In the penultimate section provides the knowledge and skills that student in the exercise of professional practice use, but there are missing knowledge and skills. The final part is about the results and overall evaluation of professional practice.

Keywords: .NET, XML, WinForms, XPath, HTML,C#, Visual studio 2010

Seznam použitých zkratk a symbolů

HTML	– HyperText Markup Language
XML	– Extensible Markup Language
XPath	– XML Path Language
CMS	– Content Management System
B2B	– Bussiness to Bussiness
B2C	– Bussiness to Consumer
HAP	– Html Agillity Pack

Obsah

1	Úvod	4
2	Popis zaměření firmy a pracovního zařazení	5
3	Zadané úkoly a postup jejich řešení	6
3.1	Úkoly a jejich řešení	6
3.2	Použité technologie při zpracovávání úkolů	13
4	Závěr	14
4.1	Uplatněné dovednosti	14
4.2	Získané dovednosti	14
4.3	Celkové zhodnocení praxe	14
5	Reference	15

Seznam obrázků

1	Screenshot programu pro univerzální export	12
---	--	----

Seznam výpisů zdrojového kódu

1	Ukázka XML Exportu	12
---	------------------------------	----

1 Úvod

Cílem této bakalářské práce je popsat absolvování odborné praxe ve firmě ZWA Company spol. s.r.o. . Odbornou praxi jsem si vybral místo teoretické práce z mnoha důvodů, jeden z hlavních důvodů je možnost vyzkoušet si nabyté znalosti a zkušenosti v praxi.

Tato práce je rozdělena na několik kapitol. V první z nich bude popsáno odborné zaměření firmy ZWA Company spol. s.r.o. a popis mého pracovního zařazení.

V další kapitole popíši, jaké úkoly jsem v rámci odborné praxe řešil a nastíním jejich řešení včetně použitých technologií. V této kapitole se také zmíním o případných problémech, které nastaly s řešením některých úkolů.

V předposlední kapitole se pokusím popsat znalosti a dovednosti, které jsem v rámci odborné praxe využil, ale také ty, které mi v průběhu praxe scházely.

Závěr bakalářské práce obsahuje celkové zhodnocení absolvované praxe.

2 Popis zaměření firmy a pracovního zařazení

Firma ZWA Company spol. s.r.o. [1] se zabývá primárně tvorbou a správou internetových stránek . Do toho můžeme začlenit tvorbu internetových obchodů, tvorbu unikátního designu, internetových stránek s CMS systémem, SEO optimalizaci. U internetových obchodů se jedná o typy B2B nebo typ B2B. B2B internetový obchod znamená, že se jedná primárně o obchodování mezi společnostmi. B2B znamená internetový obchod zaměřený na obchod s koncovými zákazníky. Do správy internetových stránek můžeme zařadit navádění a aktualizaci dat z externích zdrojů u internetových obchodů. Ve firmě ZWA Company spol. s.r.o. [1] jsem byl zařazen do funkce programátora.

3 Zadané úkoly a postup jejich řešení

Mezi první úkoly, které jsem ve firmě dostal, bylo seznámit se s firemními postupy. Jelikož jsem věděl, že mě bude čekat práce s exporty na internetové obchody zákazníků, bylo mým úkolem seznámit se s formátem XML [2] vstupů. Dále jsem se také seznámil s tím, jak se tyto vstupy mění mezi tím, než se exportují do databáze, která je napojená na internetový obchod.

3.1 Úkoly a jejich řešení

Úkoly, které jsem v průběhu praxe dostal, byly primárně zaměřeny na extrakci dat z internetových stránek. Dodavatelé produktů, které se naváděly na stránky zákazníků, neměli přístup k exportu dat z jejich internetových stránek. Z tohoto důvodu jsem byl nucen vytvořit aplikace, které tyto data obstarají. Výhodou tohoto řešení je, že se nemusí vytvářet pro každý internetový obchod specifikace XML [2] exportu.

3.1.1 Extrakce dat z internetových stránek dodavatele

Mým úkolem bylo vytvořit aplikaci, která se přihlásí na internetové stránky dodavatele produktů, nalezne všechny potřebné odkazy na produkty, které jsou k dispozici pro prodej. Dále k těmto produktům nalezne všechny žádané informace, jako například druh výrobku, název, cenu, kód, velikosti, popis a mnoho dalších informací potřebných k vystavení zboží na internetový obchod, spravovaný firmou ZWA Company s.r.o. [1]

Tento úkol jsem zpracovával v programovacím jazyce C# [3], který je součástí frameworku .NET [4]. U tohoto úkolu bylo zapotřebí vyřešit přihlášení uživatele, aby se mohla data stahovat. Toho jsem docílil použitím třídy, která dokázala udržovat cookies (jedná se o upravenou třídu WebClient), tudíž bylo možné procházet internetové stránky dodavatele tak, aby se aplikace nemusela pro každou stránku přihlašovat. Přihlášení bylo v tomto případě nutné, jelikož internetová strana dodavatele nemá veřejně dostupný internetový obchod s dodávanými produkty, tyto produkty se objeví až po přihlášení. Samotné přihlášení jsem vyřešil již zmiňovanou třídou, avšak dále jsem musel zjistit, jaké hodnoty se odesílají při přihlašování a na jakou stránku se tyto hodnoty zasílají. Potřebné informace jsem získal za pomoci internetového prohlížeče Firefox s aplikací Firebug [5], a to takovým způsobem, že jsem si spustil sledování sítě a přihlásil jsem se. Po tomto přihlášení jsem v logu zjistil, jaká data se zasílají v metodě POST a na jakou stránku. V tomto případě se jednalo o heslo a přihlašovací jméno, které se zasílalo na stránku s přihlášením. Po zjištění všech těchto informací jsem potřebná data nahrál na danou stránku a vytvořil potřebné přihlášení.

Dalším krokem při zpracování tohoto úkolu bylo vytvořit strategii, jakou se budou produkty na stránce vyhledávat. Při prvním pohledu na tuto stránku jsem zjistil, že má podobnou strukturu jako obyčejné internetové obchody. Na rozdíl od ostatních internetových obchodů měla tato společnost stránku, na které se nacházejí veškeré produkty, které jsou právě k dispozici (skladem). Této speciální záložky jsem využil takovým způsobem, že jsem si tuto stránku stáhl jako řetězec a ten jsem uložil do paměti. Poté co jsem měl

stránku uloženou v paměti, bylo nutné zjistit, jak z uložené stránky vyextrahuji potřebné odkazy na produkty, které jsou právě k dispozici. Na základě studia dostupných technologií a požadavků firmy se mi jeví vhodné použít jazyk XPath [6], se kterým jsem se seznámil při studiu na vysoké škole. Pro zpracování dat za pomoci XPath [6] dotazů existuje spousta knihoven a tříd. Po nastudování názorů na jednotlivé knihovny jsem došel k závěru, že použiji Html Agility Pack [7], což je knihovna používaná k vyhledávání a upravování pomocí jazyka XPath [6].

Po přidání knihovny Html Agility Pack [7] do mého projektu jsem musel zjistit, jakým způsobem ze stažené stránky vybrat pouze ty odkazy, které potřebuji dále zpracovávat. Abych toto zajistil, bylo nutné nastudovat si jakým způsobem správně pracovat s jazykem XPath [6] a Html Agility Pack [7]. Při vytváření správného XPath [6] dotazu jsem využil již dříve zmiňovaného Firebugu [5], který umožňuje nejen sledování sítě, ale také zobrazení kódu stránky a vyhledávání jednotlivých elementů stránky pomocí kliknutí do vizuální podoby stránky. Tímto způsobem jsem si vyhledal kolekci uzlů, které obsahovaly v attributech jednotlivé odkazy na produkty. Tyto odkazy jsem si uložil a následně stáhl. V tomto bodě mám již staženy HTML kódy stránek s jednotlivými produkty. Nyní bylo nutné z těchto zdrojových kódů vyextrahovat potřebná data a to název produktu, výrobce, cenu, velikost. Dále pak skupinu do jaké produkt spadá a v neposlední řadě také odkaz na obrázek. Pro každou jednu z těchto informací bylo zapotřebí vytvořit XPath [6] dotaz za pomoci Firebug [5] a tento dotaz s pomocí Html Agility Pack [7] zakomponovat do zdrojového kódu aplikace. Po vytvoření veškerých dotazů bylo nutné pracovat s vyextrahovanými daty. Tato práce spočívala v tomto případě v převedení ze zahraniční měny na měnu českou a přidání marže. Po zdárném dokončení této části přišly na řadu obrázky. Obrázky jsem v průběhu zpracování jednotlivých stránek ukládal do slovníku, ve kterém se nalézaly informace o názvu a adrese jednotlivých obrázků. Pro stažení všech obrázků bylo nezbytné projít celý slovník a stáhnout jednotlivé obrázky. Jako jeden z posledních kroků bylo nutné zrychlit zpracování úkolu, toho jsem docílil vláknovým zpracováním, a to tak, že jsem si jednotlivé linky rozdělil na několik částí a tyto části se postupně zpracovávaly. Poslední část toho úkolu spočívala v testování. V průběhu testování jsem musel zkontrolovat, zdali vše pracuje jak má a jestli jsou ošetřeny všechny chyby, které mohou nastat při zpracovávání.

3.1.2 Tvorba XML exportu ze zahraničního internetového obchodu, který neposkytuje XML export

U druhého typu úkolu se jednalo o vytvoření kompletního exportu ze zahraničního internetového obchodu, který je denně aktualizován. Protože tento internetový obchod neposkytoval svůj vlastní export, bylo nutné export vytvořit. V exportu měly být informace o produktu jako například cena, dostupné velikosti, název zboží a obrázek tohoto zboží. V tomto případě, jelikož se jednalo o internetový obchod s několika tisíci druhy produktů, jsem musel detaily spolupráce řešit osobně s majitelem tohoto internetového obchodu.

U tohoto úkolu, stejně jako u předcházejícího, bylo překážkou přihlášení do internetového obchodu. K tomu, abych se přihlásil a mohl stahovat potřebné informace, jsem použil již nabyté zkušenosti z předchozího projektu. Využil jsem již zmiňované třídy, která si pamatuje cookies. Toto bylo potřebné k tomu, abych mohl procházet a stahovat jednotlivé stránky z tohoto internetového obchodu. Postup byl následující: ve Firebug [5] jsem spustil sledování sítě a zachovávání informací. Po přihlášení jsem zjistil, že v metodě POST jsou tři parametry, které je potřebné nahrát při přihlašování na stránku vůči které proběhlo přihlášení. V těchto parametrech byly informace o přihlašovacím jméně, hesle a parametrem, který určoval, že se jedná o přihlašování.

Po úspěšném přihlášení skrze vytvářenou aplikaci bylo mým úkolem analyzovat, jakým způsobem se bude tento internetový obchod procházet. Jelikož se jedná o internetový obchod s tradičním uspořádáním, strategie byla taková, že prvně se zpracují linky v hlavním menu a následně pak jednotlivé produkty. První věc na kterou jsem se zaměřil byla extrakce všech odkazů z hlavního menu. Tyto odkazy jsem vyhledával pomocí XPath [6], který jsem do C# [3] zakomponoval za pomoci Html Agility Pack [7]. K tomu, abych vyhledal pouze ty linky, které potřebuji zpracovat, bylo nutné jednoznačně definovat o jaké uzly se jedná.

Velkým pomocníkem při řešení toho, jak vytvořit XPath [6] dotaz pro mě byl Firebug [5] a jeho záložka HTML, ve které jsem zjišťoval, jaký je název například elementu, ve kterém je obaleno menu. Po tom co jsem vytvořil XPath [6] dotaz, který byl funkční, bylo nutné stáhnout jednotlivé zdrojové kódy. Tyto zdrojové kódy se stahují na základě odkazů, které se ukládají do seznamu za pomoci vytvořeného XPath [6] dotazu. Po stažení všech zdrojových kódů, jsem musel zajistit také to, aby se stáhly tyto kódy kompletní. Tímto je myšleno, že po otevření jednoho odkazu z hlavního menu se načte stránka, která obsahuje i stránkování, což v mém případě znamenalo to, že je nutné zjistit, jakým způsobem je sestaven odkaz, když se přehází v rámci stránkování.

Tento problém jsem vyřešil zvolením strategie, která pro jednotlivé odkazy zjistí počet stránek ve stránkování, což znamená zjistit poslední stranu daného stránkování a rozsah mezi první a poslední stránkou doplnit pomocí cyklu. Tento cyklus sestavil odkaz tak, že přidal k základní adrese text, potřebný k tomu, aby se změnila stránka a doplnil i číslo námi požadované stránky. Takto sestavený odkaz se přidal do seznamu. Po provedení stránkování na všech stranách hlavního menu jsem pomocí uložených adres stáhl jejich zdrojové kódy a přidal je k již staženým kódům.

Nyní jsou staženy všechny stránky obsahující jednotlivé produkty na stránkách dodavatele. Následuje extrahování odkazů vedoucích k jednotlivým produktům. Zde jsem použil již dříve zmiňované vyhledávání pomocí Html Agility Pack [7], který využívá XPath [6], což znamená, že pro každou staženou stranu je provedeno vyextrahování odkazů vedoucích na jednotlivé produkty. Tyto odkazy jsou uloženy do seznamu. Po vyextrahování všech odkazů je nutné stažení obsahu každého odkazu v seznamu. Provádění této části je velice časově náročné, protože v tomto případě se jedná o tisíce druhů produktů (odkazů), které je nutné stáhnout.

Po dokončení stahování zdrojových kódů všech produktů je potřebné každý zdrojový kód projít a vyextrahovat z něj potřebná data. Data se vyhledávají za pomoci XPath [6] dotazů, kdy u každého dotazu můžeme zjistit veškeré vlastnosti daného uzlu, jako jsou například vnitřní text, informace o jaký element se jedná, jestli má rodiče, kdo je předek a jaké má atributy. Veškerá potřebná data se vyextrahují a zapíší do řetězce, ze kterého se následně vytvoří XML [2] dokument určený pro export do firemního systému. Při tomto zpracovávání dochází také k vyhledávání obrázků. Názvy těchto obrázků a jejich adresy jsou zapisovány do slovníku. Tento slovník je po dokončení extrahování dat postupně procházen a obrázky jsou stahovány a ukládány.

Po otestování aplikace, zda pracuje správně, došlo k úpravě aplikace tak, aby se data zpracovávala pomocí vláken. Toho zpracování se využívá zejména u zpracování jednotlivých produktů. Výsledkem na konci zpracování je XML [2] export, ve kterém se nacházejí veškeré informace o produktech nalezených na internetové straně dodavatele včetně názvů obrázků, které jsou samozřejmě také staženy.

3.1.3 Vytvoření univerzální aplikace pro vytváření exportů z internetových stránek.

Zde bylo mým úkolem vytvořit aplikaci, která bude zpracovávat informace z internetových stránek a bude mít univerzální použití. Při prvním pohledu na toto zadání bylo jasné, že jeden z úkolů, které mě čekají je zjistit, čím jsou si jednotlivé internetové obchody podobné. Po analýze několika internetových obchodů jsem dospěl k závěru, že většina z nich je postavená na principu hlavního menu. Po kliknutí na některý odkaz se otevře v hlavní části stránky nabídka právě vybraného typu zboží. Po detailnějším studiu těchto stránek jsem zjistil, že stránka s nabídkou jednotlivých druhů zboží obsahuje také většinou stránkování.

Pro nastavování parametrů jsem použil grafické rozhraní, které obsahuje textová pole pro zadání adresy stránky, XPath [6] hlavního menu, XPath [6] jednotlivých položek, XPath [6] pro stránkování a XPath [6] pro obrázky. V grafickém rozhraní se také nachází zaškrťovací políčko, které udává, jestli má být k linku vložena i základní adresa. Poslední element nacházející se na grafickém rozhraní aplikace je tabulka pro zadání názvu pro export a XPath [6] jednotlivých informací, které budeme vybírat ze stránky s produktem. Dole jsou tři tlačítka, která jsou potřebná k uložení konfigurace, načtení konfigurace a provedení nastaveného exportu.

Při vytváření této aplikace jsem postupoval obdobně jako v předchozích případech. V první řadě bylo nutné vyextrahovat linky z hlavního menu a to tím způsobem, že se stáhne zdrojový kód základní adresy a na tento kód je použit XPath [6] dotaz zadaný uživatelem. Tyto linky jsou v zásadě dva typy. Jeden typ je odkaz kompletní, což znamená, že obsahuje i základní adresu stránky. Druhý typ obsahuje pouze to, co je za základní adresou stránky. U druhého typu odkazu je potřebné (ještě před tím, než dojde ke stáhnutí linku) přidat základní adresu před daný odkaz. Toto se provádí takovým způsobem, že algoritmus porovná, jestli základní adresa nebo část, která se bude k adrese přidávat, obsahuje lomítko nebo ne. Na základě toho porovnání se vytvoří adresa, která je přidána do seznamu, který bude stahovat zdrojové kódy dalších stránek.

Po provedení toho algoritmu na všechny odkazy následuje stažení všech zdrojových kódů. V tomto bodě již máme staženy zdrojové kódy všech hlavních kategorií. Následuje procházení všech těchto zdrojových kódů a hledání, zdali obsahuje stránkování na základě XPath [6] dotazu, který zadal uživatel. Pokud je nalezeno stránkování, dochází k vyhledávání počtu stránek, které jsou stránkovány. Na základě údajů o počtu stránek se provede cyklus, který vytvoří adresy pro stránky, které zatím nebyly staženy, což znamená, jestliže strana obsahuje stránkování s deseti stránkami je potřeba vytvořit adresy pro dalších devět stránek a ty následně stáhnout.

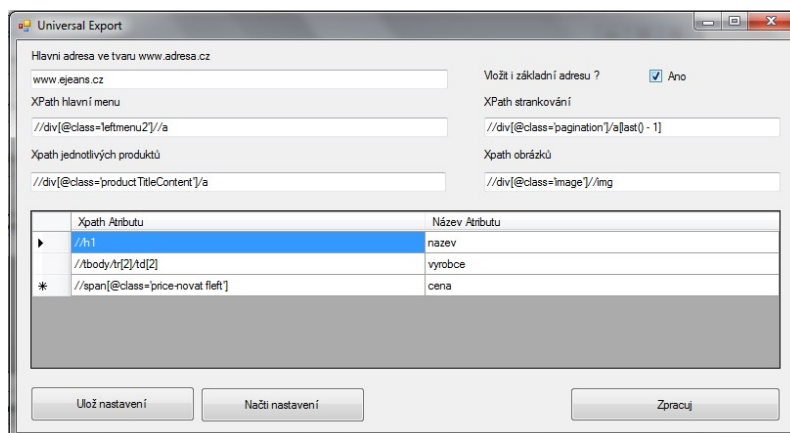
V tomto bodě při testování došlo k prvnímu problému, který jsem nepředpokládal. Problém byl v tom, že některé internetové obchody nemají viditelné kompletní stránkování. Jinými slovy to znamená, že například stránkování je na dvaceti stránkách, bohužel ne všechny jsou viditelné. Kdyby byla viditelná poslední stránka, k problémům by nedocházelo, avšak pokud jde vidět jen prvních deset a až po kliknutí na některý z druhé poloviny číselníku se řada posune na další čísla, tím nastává problém. Tento problém jsem vyřešil za pomoci procedury, která zkontroluje poslední stránku v řadě stránkování a zkusí použít XPath [6] pro stránkování. Jestli jsou XPath [6] stejné, pak z toho vyplývá,

že i stránky jsou stejné, takže další stránky nenásledují. Pokud XPath [6] stejné nejsou, pak jsou stránky rozdílné a musí se pokračovat v zjišťování, kde je konec této řady.

Po stažení veškerých stránek obsahujících odkazy na všechny produkty bylo zapotřebí vyextrahovat všechny odkazy vedoucí k jednotlivým produktům. Tyto odkazy se hledaly na základě řetězce, který zadal uživatel jako XPath [6] pro jednotlivé produkty. Po vyhledání těchto adres se stáhly zdrojové kódy stránek jednotlivých produktů. Nad těmito stránkami se použil takový počet XPath [6] dotazů, kolik je zadáno atributů v tabulce. Při zpracování se také k jednotlivým produktům přidával název obrázku a odkaz na něj. Po dokončení tohoto algoritmu se následně stahují obrázky a to tak, že ke každému produktu, který obrázek obsahuje je přiložen název obrázku stejný, jako je název stahovaného obrázku. Výhodou toho řešení je to, že názvy obrázků již nemusíme žádným způsobem upravovat.

Po dokončení základní verze jsem jsem začal pracovat na verzi, která bude obsahovat vláknové zpracování. Pro tento druh zpracování jsem si musel rozdělit data tak, aby byla zpracovatelná odděleně. Celá aplikace je napsaná takto. Jako první se zpracuje hlavní stránka, což znamená stáhnutí HTML kódu a extrakce jednotlivých odkazů z hlavního menu. Po tomto kroku nastává rozdělení odkazů na části. Části jsou dvě nebo čtyři. V každé z těchto částí se provádí vláknové zpracování. Následující krok je stáhnutí zdrojových kódu stránek, na které vedou odkazy z hlavního menu. Po té, co jsou všechny HTML kódy staženy, dochází ke sloučení všech zdrojových kódů do jednoho seznamu. Tento krok je nezbytný k rovnoměrnému rozložení počtu kódů v jednotlivých seznámech. Rozložení seznamu se provádí na základě počtu záznamů v seznamu, avšak největší počet záznamů v jednom seznamu je pět set. Poté jsou seznamy postupně zpracovány. Již zpracované seznamy se uvolňují, aby nezabíraly místo v paměti. Zpracování probíhá v maximálně třech vláknech. Po skončení zpracování se všechny výsledky spojí do jednoho řetězce. Tento řetězec je obalen do elementu s označením export a jako atribut obsahuje datum vytvoření exportu.

Výsledkem zpracování dat je XML [2] soubor s výstupem, jaký si uživatel navo-lil v konfiguraci za pomocí XPath [6] dotazů. Pokud si uživatel nastaví XPath [6] i pro obrázky, pak součástí exportu je také složka s obrázky. Aplikace je i nadále vyvíjena.



Obrázek 1: Screenshot programu pro univerzální export

```
<Export date='8.3.2012_12:13:01'>
  <Item id='1'>
    <název>dámské kalhoty jeans LEE – BONNIE</název>
    <výrobce>Lee</výrobce>
    <cena>1 400 Kč (56 EUR)</cena>
    <img>_vyr_295damske–kalhoty–jeans–LEE–BONNIE–L302ALMO1.jpg</img>
  </Item>
  <Item id='2'>
    <název>dámské kalhoty jeans LEE – CAMERON</název>
    <výrobce>Lee</výrobce>
    <cena>1 167 Kč (47 EUR)</cena>
    <img>_vyr_77damske–kalhoty–LEE–L349INCW2.jpg</img>
  </Item>
  <Item id='3'>
    <název>dámské kalhoty jeans LEE – CAMERON</název>
    <výrobce>LEE</výrobce>
    <cena>1 833 Kč (73 EUR)</cena>
    <img>_vyr_306damske–kalhoty–jeans–LEE–CAMERON–L349AGNC3.jpg</img>
  </Item>
  <Item id='4'>
    <název>dámské kalhoty jeans LEE – CAMERON</název>
    <výrobce>Lee</výrobce>
    <cena>1 583 Kč (63 EUR)</cena>
    <img>_vyr_390damske–kalhoty–Lee–L349OFKN4.jpg</img>
  </Item>
  <Item id='5'>
    <název>dámské kalhoty jeans LEE – CAMERON DARK STONE</název>
    <výrobce>Lee</výrobce>
    <cena>1 167 Kč (47 EUR)</cena>
    <img>_vyr_31L349INBS5.jpg</img>
  </Item>
</Export>
```

Výpis 1: Ukázka XML Exportu

3.2 Použité technologie při zpracovávání úkolů

3.2.1 Html Agility Pack

Html Agility Pack (HAP) [7] je HTML parser, který si dokáže poradit s běžně se vyskytujícími HTML kódy tak i s kódy, které nejsou úplně validní. Při práci s touto knihovnou je potřebné znát alespoň základy XPath [6] jazyka, jelikož pro adresování jednotlivých uzlů se používají XPath [6] dotazy. HAP je dostupný jako .NET [4] knihovna. Tato knihovna byla mým velkým pomocníkem při řešení mých úkolů, protože při správném sestavení XPath [6] dotazu se v proměnné nacházely veškeré informace o daném uzlu. Tyto informace například obsahují veškeré atributy daného uzlu (tagu), text, který uzel obsahuje a také vnější HTML.

3.2.2 Firebug

Firebug [5] je rozšíření internetového prohlížeče Firefox, které slouží v obecném případě k ladění internetových stránek. V mém případě jsem Firebug [5] použil, protože bylo nutné zjišťovat jaká data jsou předávána při přihlašování v metodě POST, abych tyto informace využil při přípravě algoritmu pro přihlašování. Další důležitou roli Firebug [5] sehrál jako pomocník při zkoumání HTML kódu a vyhledávání správného XPath [6] dotazu, protože tento doplněk umožňuje najít část HTML kódu po kliknutí na prvek v grafické podobě internetové stránky.

3.2.3 XML

XML [2] je značkovací jazyk, určený k výměně dat mezi aplikacemi. V mém případě šlo o vytvoření XML [2] souboru, který byl poté importován do systému, který je určen pro navádění dat na internetové obchody klientů.

4 Závěr

4.1 Uplatněné dovednosti

Znalosti uplatněné při mé odborné praxi byly zejména z oblasti programovacích jazyků a to především z objektově orientovaného jazyka C# [3] vyučovaného v předmětu Programovací jazyky II. Tato znalost mě ušetřila spoustu času, který jsem nemusel strávit při studování tohoto programovacího jazyka. Dále jsem využil znalostí z předmětu Vývoj internetových aplikací, ze kterého jsem uplatnil znalost jazyka XPath [6]. V neposlední řadě jsem při vytváření grafického rozhraní využil znalosti nabyté v předmětu Uživatelská rozhraní.

4.2 Získané dovednosti

Dovednosti, které jsem získal během své praxe ve společnosti ZWA Company spol. s r.o. byly především komunikace s lidmi ve firmě i s lidmi mimo firmu. Dále jsem si prohloubil znalosti z programovacích jazyků a to především z programovacího jazyka C# [3], jako například práce s komponentami grafického rozhraní a vláknové zpracování. Také díky tomu, že jsem při vyhledávání používal jazyk XPath [6] zlepšil jsem si znalost tohoto adresovacího jazyka především v tom, že jsem se naučil adresovat uzly velice přesně. Za další velkou zkušenost považuji to, že jsem se naučil používat Html Agility Pack [7].

4.3 Celkové zhodnocení praxe

Absolvování odborné praxe беру jako velký přínos, protože ve firmě ZWA Company spol. s r.o. [1] jsem získal mnoho nových znalostí a zkušeností, mezi které mohu zařadit komunikaci s lidmi a vytváření aplikací, které mají reálné využití, jako je například aplikace pro univerzální export, kterou firma bude i nadále používat. Nabyté znalosti a zkušenosti využiji v dalším studiu a také v dalším zaměstnání. Možnost zpracovat bakalářskou práci ve formě absolvování bakalářské praxe mě připadá více přínosná než klasická forma bakalářské práce, protože během absolvování bakalářské praxe se student naučí spoustu věcí z reálného chodu společnosti.

5 Reference

- [1] ZWA Company. *Tvorba a optimalizace e-shopů s originálním designem* [online]. [cit. 2012-03-24].
Dostupné z: <http://www.zwa.cz>
- [2] Jiří Kosek. *XML pro každého* [online]. [cit. 2012-03-24].
Dostupné z: <http://www.root.cz/knihy/xml-pro-kazdeho>
- [3] John Sharp. *Microsoft Visual C# 2010* COMPUTER PRESS, 2010
- [4] Jeff Prosise. *Programování v Microsoft .NET* COMPUTER PRESS, 2003
- [5] Firebug. *Documentation* [online]. [cit. 2012-03-24].
Dostupné z: <http://www.getfirebug.com>
- [6] W3C School. *Xpath Tutorial* [online]. [cit. 2012-03-24].
Dostupné z: <http://www.w3schools.com/xpath>
- [7] Html Agility Pack. *Documentation, Examples* [online]. [cit. 2012-03-24].
Dostupné z: <http://htmlagilitypack.codeplex.com>